PROJECT REPORT
ON

BLACK JACK GAME

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING - 128

BATCH- E6

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

SOFTWARE DEVELOPMENT FUNDAMENTALS
LAB-2

TEAM MEMBER NAMES	ENROLLMENT NUMBER
HARSHIT SRIVASTAVA	**9922102107**
VINAY BHARDWAJ	**9922102101**
NEWTON RAJ	**9922102104**

Introduction:

The Black Jack game is a popular card game played between a player and a dealer. The objective of the game is to reach a card value as close to 21 as possible without exceeding it. This project aims to develop a Black Jack game using Object-Oriented Programming (OOP) concepts in the C++ programming language. The game will be implemented using various modules and will incorporate features of C++ to enhance functionality and user experience.

Problem Statement :

The objective of this project is to implement a Black Jack game using OOP in C++. The game should allow a player to play against a dealer, following the rules of Black Jack. The project should fulfill the following requirements:

a) Create a shuffled deck of cards and deal cards to the player and dealer.
b) Implement the game logic, including calculating card values, checking for wins or losses, and handling player decisions.
c) Allow the player to request additional cards or stand based on their hand value.
d) Implement the dealer's decision-making process based on predefined rules.
e) Determine the winner of each round and keep track of the player's score.
f) Provide a user-friendly interface to display the game status and prompt player actions.

Modules:

The project consists of the following modules:

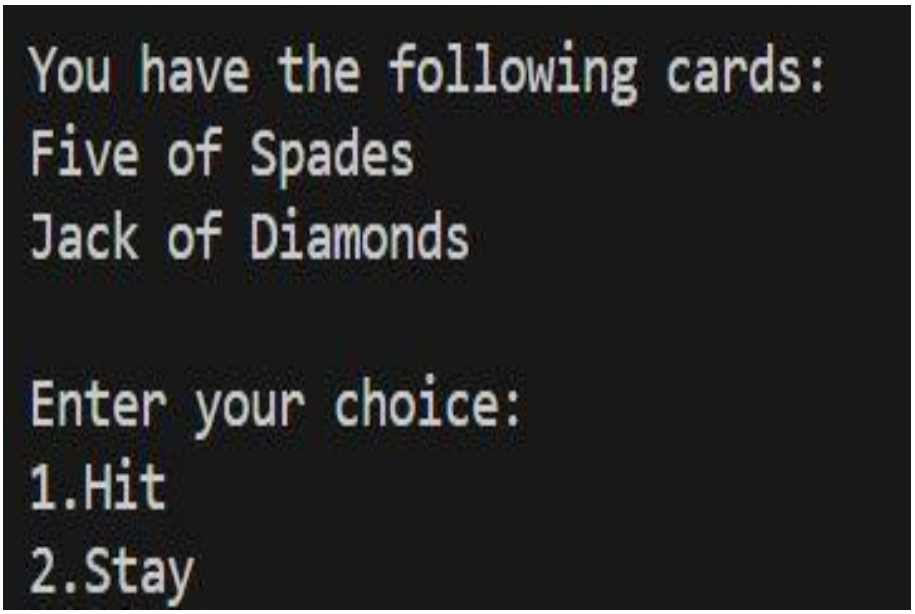
a) Deck Module: This module is responsible for creating and managing the deck of cards used in the game. It will generate a shuffled deck, deal cards to the player and dealer, and handle card drawing during the game.

b) Player Module: This module represents the player in the game. It will handle actions such as requesting cards, calculating the total card value, and managing the player's hand.

c) Dealer Module: This module represents the dealer in the game. It will manage the dealer's hand, handle actions such as drawing cards, and make decisions based on the game rules.

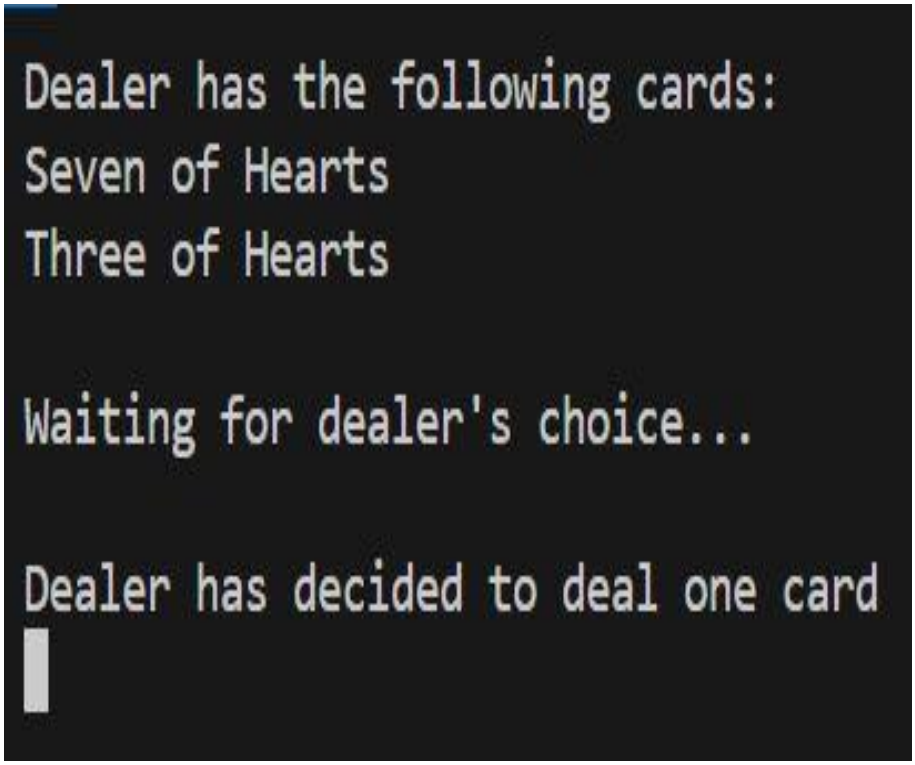
d) Game Module: This module acts as the main controller of the game. It will coordinate the interactions between the player, dealer, and deck modules. It will handle game logic, such as determining the winner, managing the game flow, and providing a user interface.

OUTPUT:



```
You have the following cards:
Four of Spades
Eight of Diamonds

Enter your choice:
1.Hit
2.Stay
=>4

Invalid input!! Try again
```

```
You have the following cards:
Five of Spades
Jack of Diamonds
King of Spades

You have BUSTED!! Dealer WINS

Press Y to play again / Any other key to exit
```




```
You have the following cards:
Five of Hearts
Four of Clubs
Six of Hearts
Two of Spades


Dealer has the following cards:
Seven of Hearts
Three of Hearts
Eight of Diamonds


By comparison of cards:
Dealer has won the game.

Press Y to play again / Any other key to exit
=>1


END OF PROGRAM
```

Features of C++ to be used:

The Black Jack game project utilizes various features of C++ to enhance functionality and maintainability. Some of the key features used in this project are:

a) *Object-Oriented Design Principles: The project follows object-oriented design principles such as encapsulation, inheritance, and polymorphism to achieve modularity, extensibility, and code reusability.*

b) *Classes and Objects: The game is implemented using a class-based approach, with each module represented as a class and instantiated as objects.*

c) *Encapsulation: The data and methods of each class are encapsulated, allowing for better organization and separation of concerns.*

d) *Inheritance: Inheritance is used to create a base class for the player and dealer modules, as they share common attributes and behaviors.*

e) *Polymorphism: Polymorphism is employed through virtual functions and function overriding to provide different implementations for common methods in the player and dealer classes.*

f) *Operator Overloading: Operator overloading is employed to define custom behavior for operators, such as comparison operators (=) to store values in private variables suit and rank.*

g) Dynamic Memory Allocation: Dynamic memory allocation is utilized to manage the creation and destruction of objects dynamically. It ensures efficient memory utilization and allows for flexibility in managing resources.

h) Error Handling: Appropriate error handling techniques are implemented, such as exception handling, to handle unexpected situations and provide meaningful error messages to the use.

CONCLUSION:

Overall, the utilization of these C++ features enhances the modularity, reusability, and readability of the Black Jack game project. It allows for better organization and structure of the code, promotes code reuse through inheritance, and provides a user-friendly interface for a smooth gaming experience.